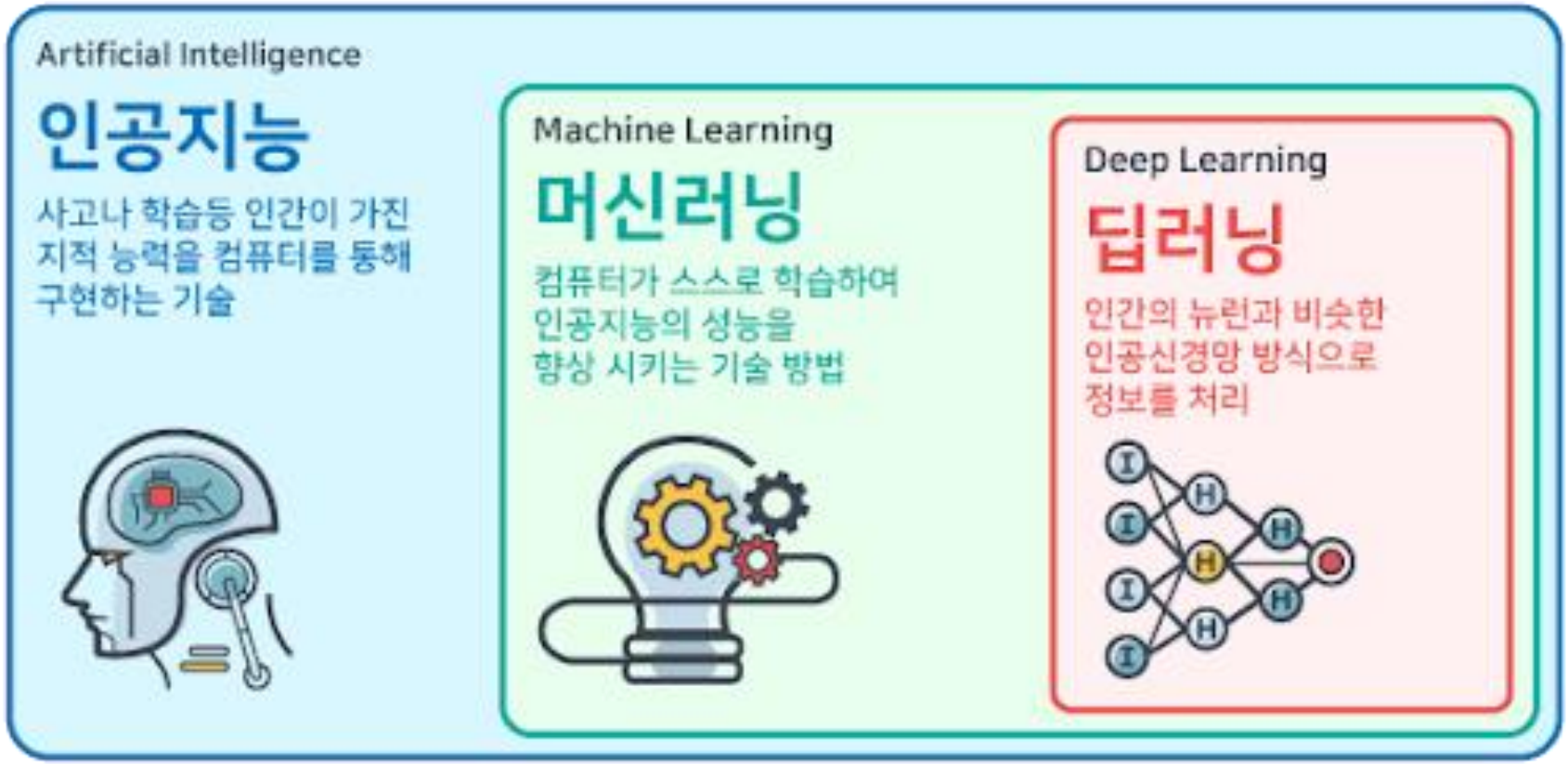
머신러닝 & 딥러닝 8

AI 학술동아리 <MLP>

- Index

- Artificial Intelligence(인공지능)



DL 라이브러리는 다른 ML 라이브러리와 다르게 GPU사용해서 ANN훈련
- GPU는 벡터와 행렬 연산에 매우 최적화

1. Sequential Data(순차 데이터)

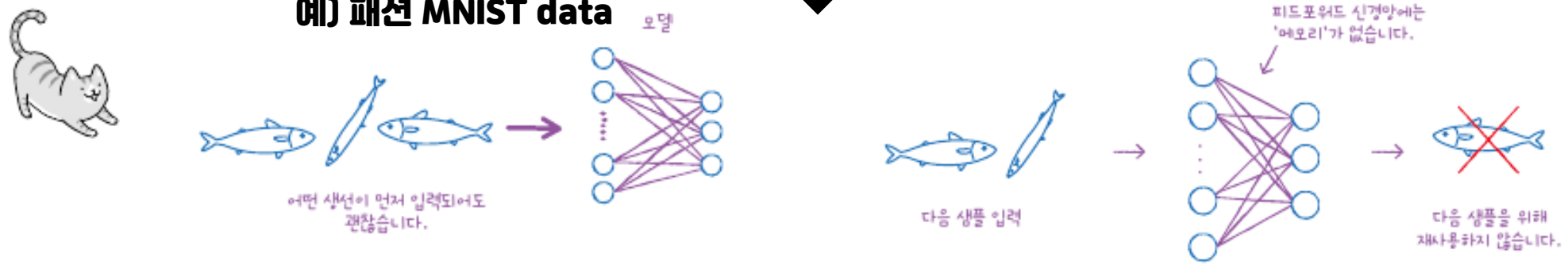
RNN은 순환 계산 수행 -> 순서를 따짐

sequential data : 순서에 의미가 있는 data

time series data(시계열 데이터) : 일정한 시간 간격으로 기록된 데이터

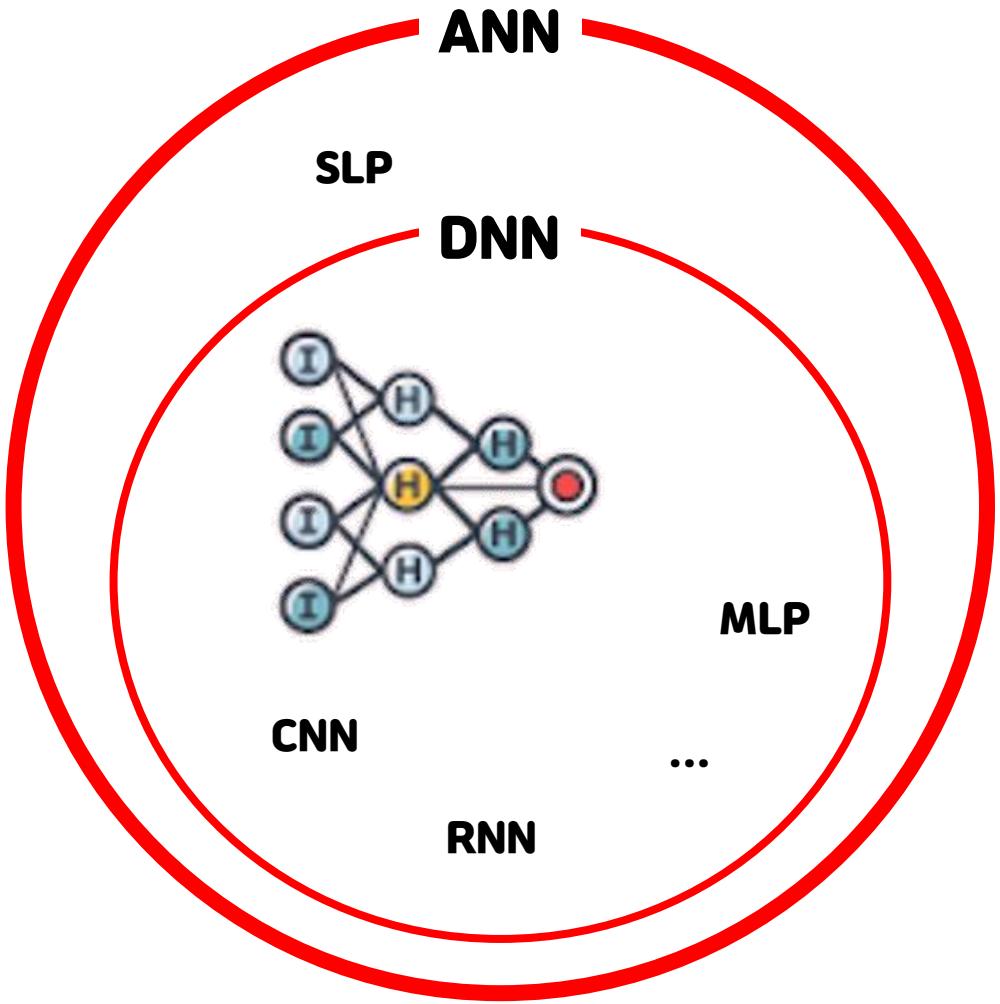
sequence : 하나의 샘플

일반 데이터 : 순서에 의미가 없는 data
예) 패션 MNIST data



MLP, CNN은 정방향 계산 수행 -> 순서를 따지지 않음
=> feedforward neural network(FFNN)(피드포워드 신경망)

- Artificial Neural Network(인공 신경망)



ANN을 줄여서 NN(Neural Network)라고도 함

ANN을 DL이라고도 함

ANN(Artificial Neural Network) = 인공 신경망

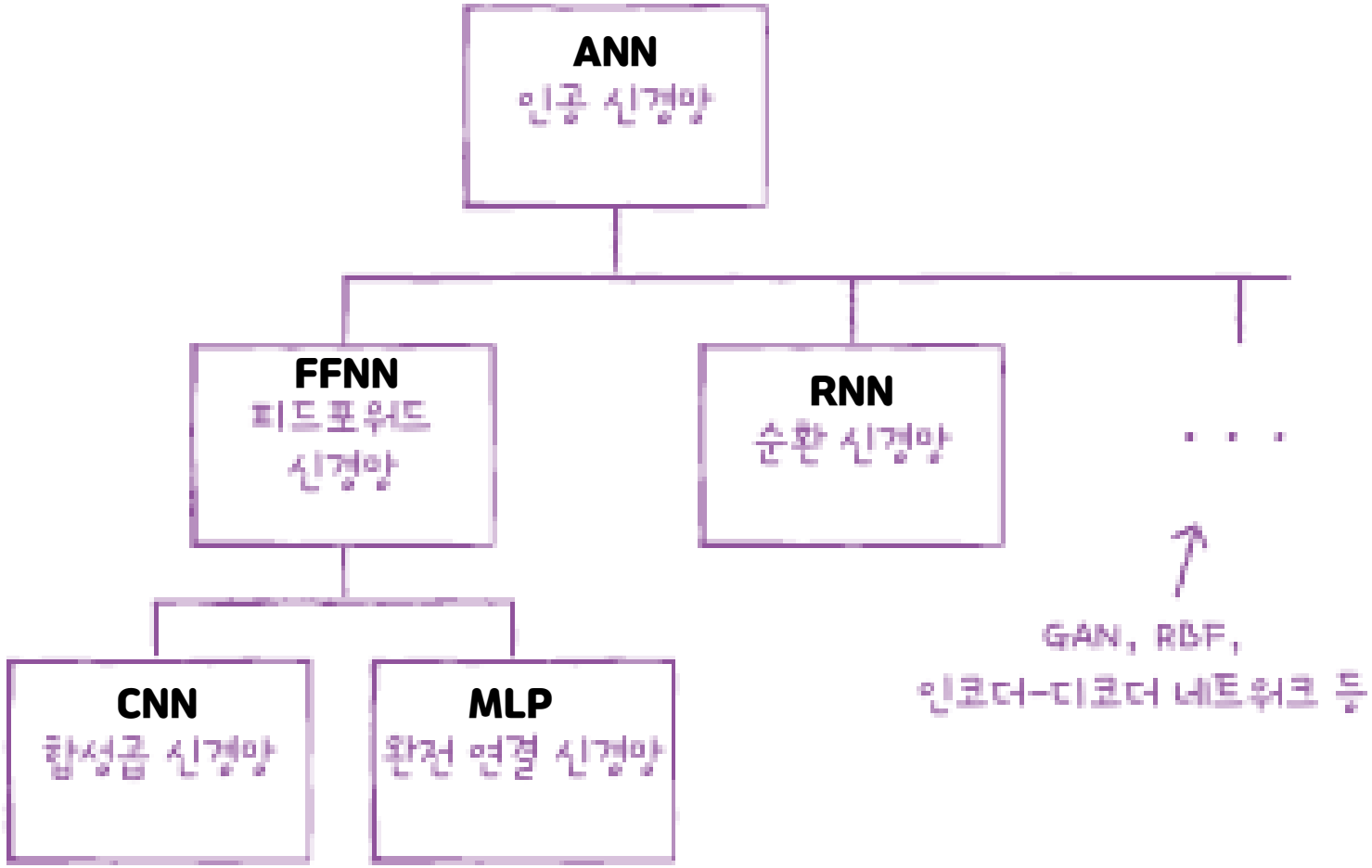
DNN(Deep Neural Network) = 심층 신경망
= Deep Learning

-> Layer가 여러개

SLP(Single Layer Perceptron) = 단층 퍼셉트론

MLP(Multi Layer Perceptron) = 다층 퍼셉트론

- ANN

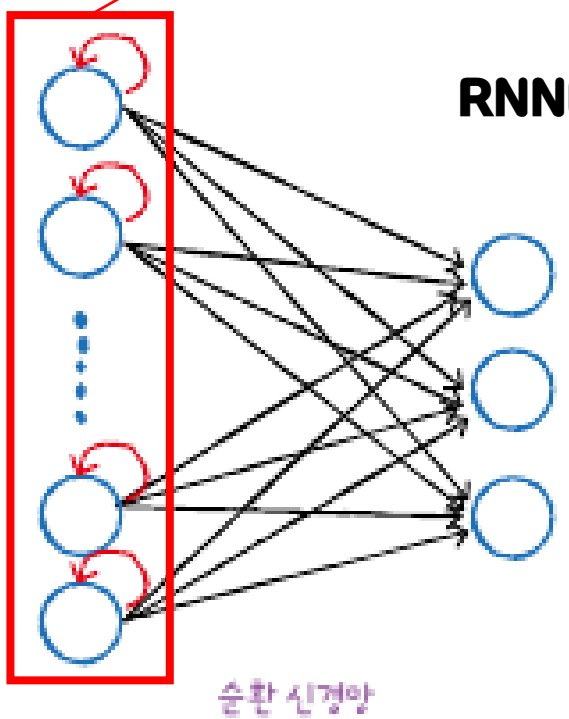

ANN
인공 신경망
FFNN
피드포워드
신경망
RNN
순환 신경망
CNN
합성곱 신경망
MLP
완전 연결 신경망
GAN, RBF,
인코더-디코더 네트워크 등

RNN은 MLP와 거의 비슷

2. Recurrent Neural Network(순환 신경망)

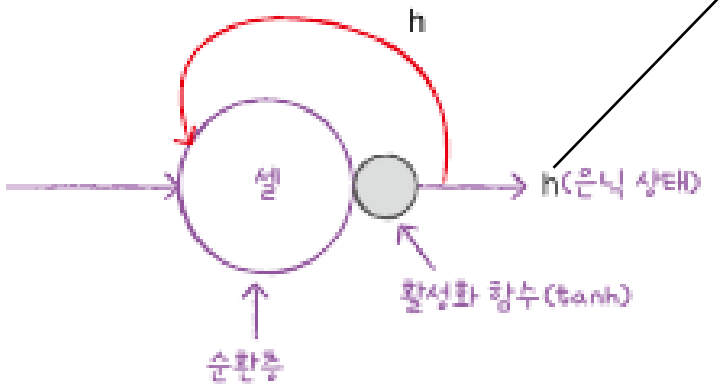
RNN은 recurrent layer을 가지고 있는 NN

cell

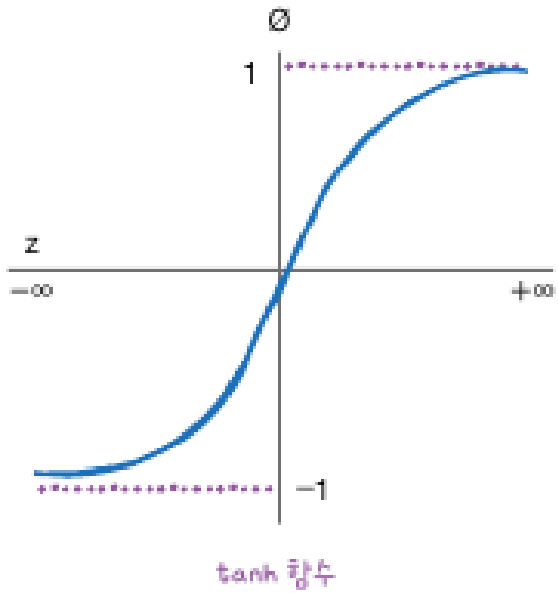


recurrent layer(순환층)

RNN에서는 특별히 layer를 cell이라고 부름



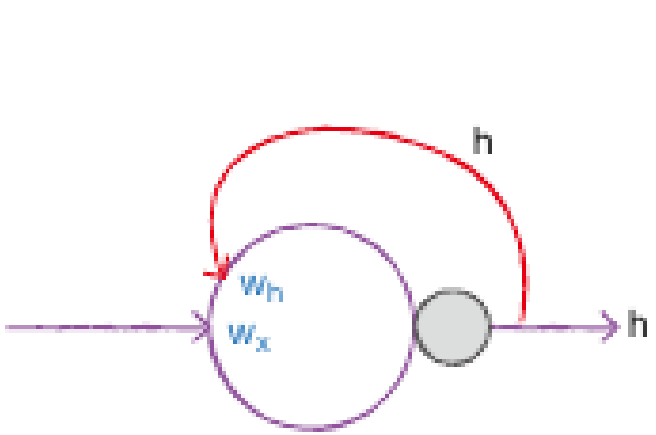
cell의 출력을 hidden state라고 부름



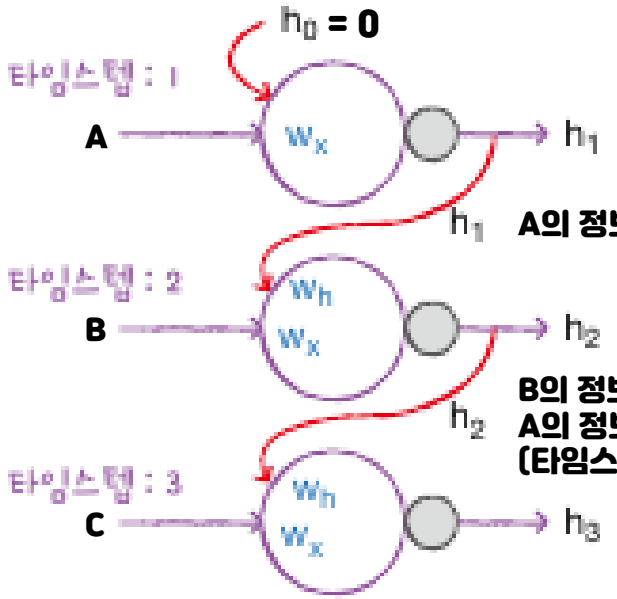
RNN에서는 보통 activation function으로 tanh를 사용

2. RNN

timestep : 샘플을 처리하는 한 단계



타임스텝으로 펼칩니다.

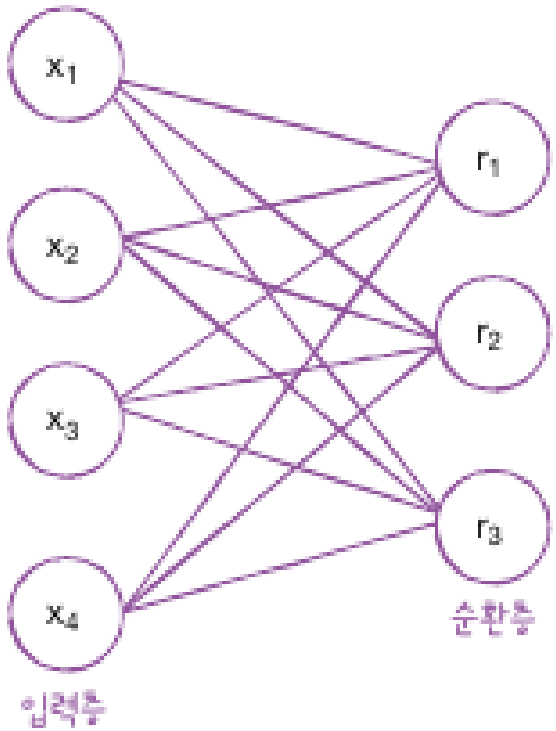


h_1 A의 정보를 담고 있음

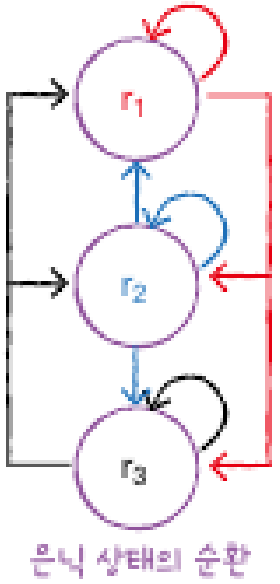
h_2 B의 정보를 담고 있음
A의 정보도 담고 있음
(타임스텝을 거치며 A의 정보는 살짝 희미해짐)

h_3 C의 정보를 담고 있음
B의 정보도 담고 있음
A의 정보도 담고 있음
(타임스텝을 거치며 B의 정보는 살짝 희미해지고,
A의 정보는 그보다 더 희미해짐)

2. RNN



+

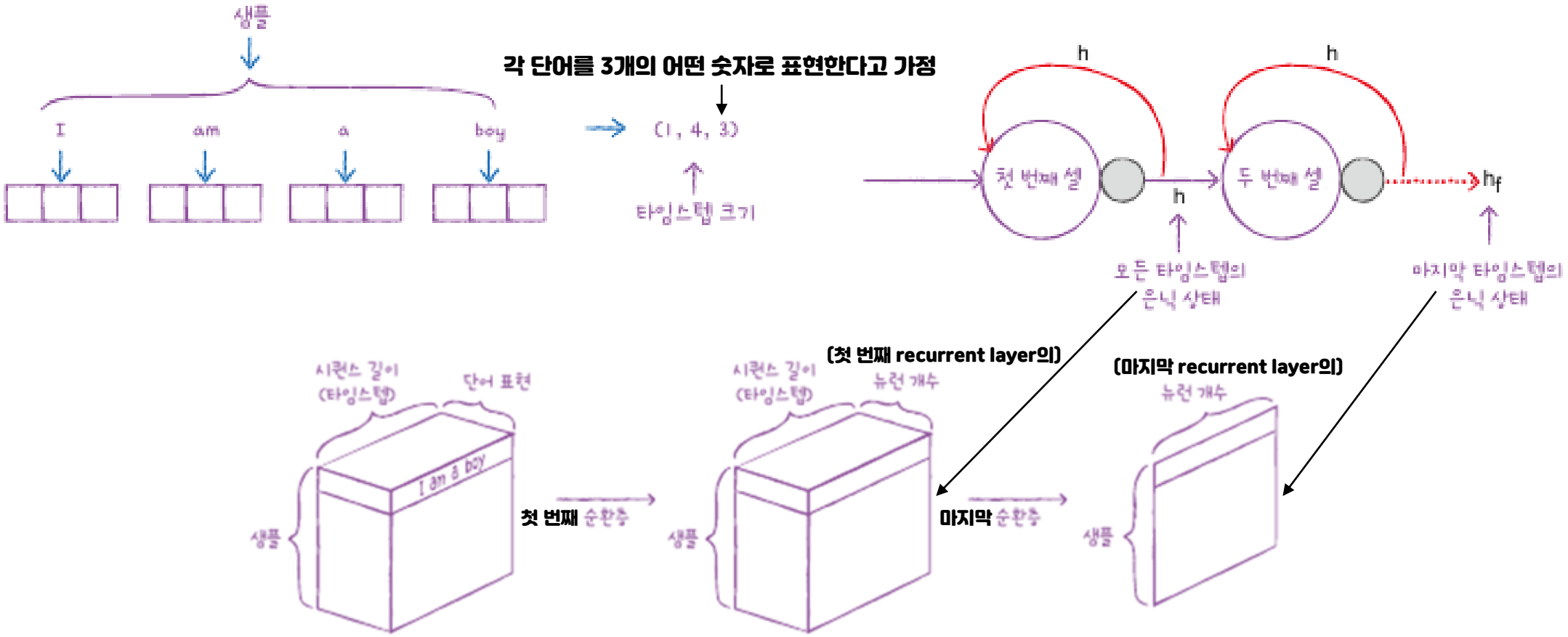


recurrent layer

+ bias(절편) = model parameter 수

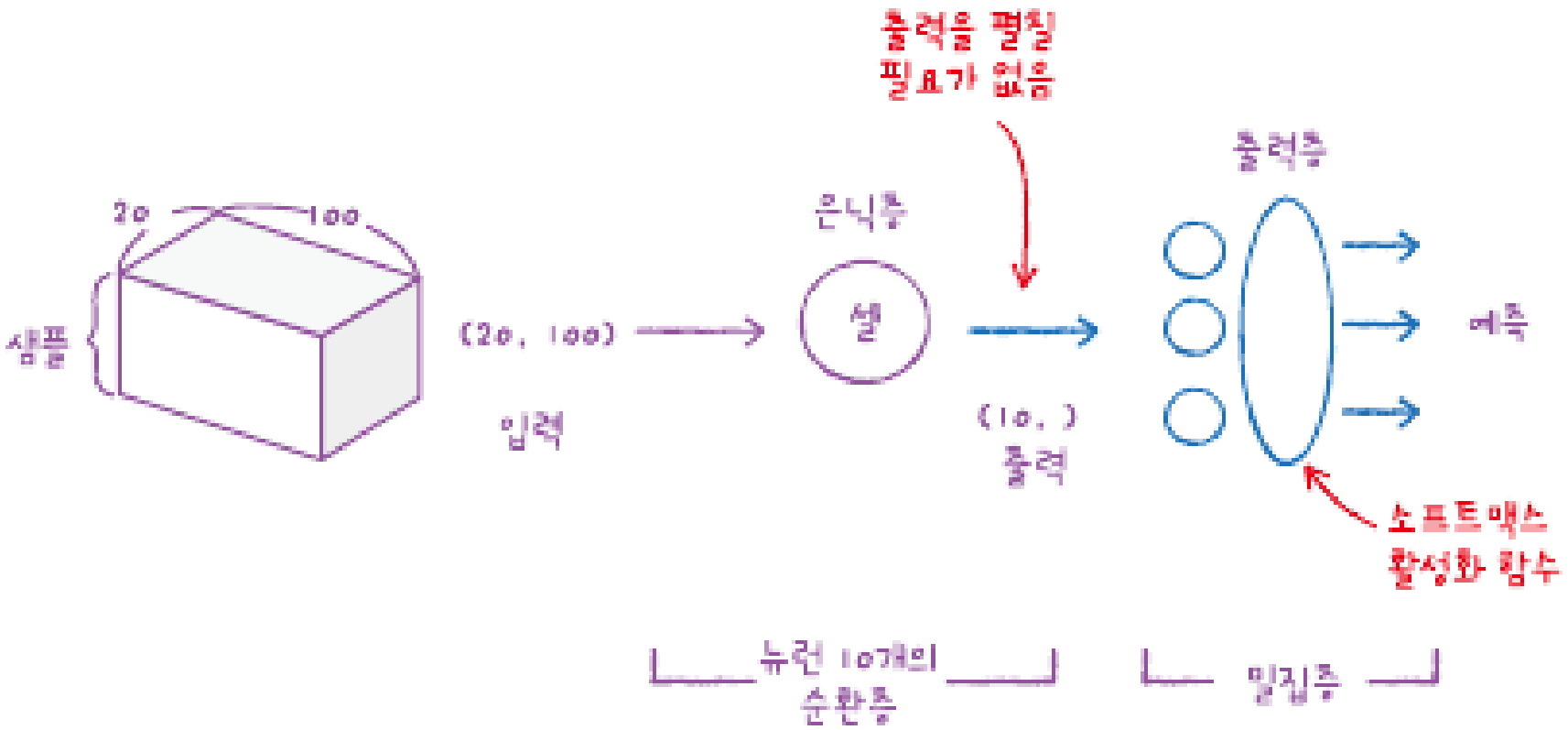
Model parameter 수 = w_x + w_h + bias = 12 + 9 + 3 = 24

2. RNN



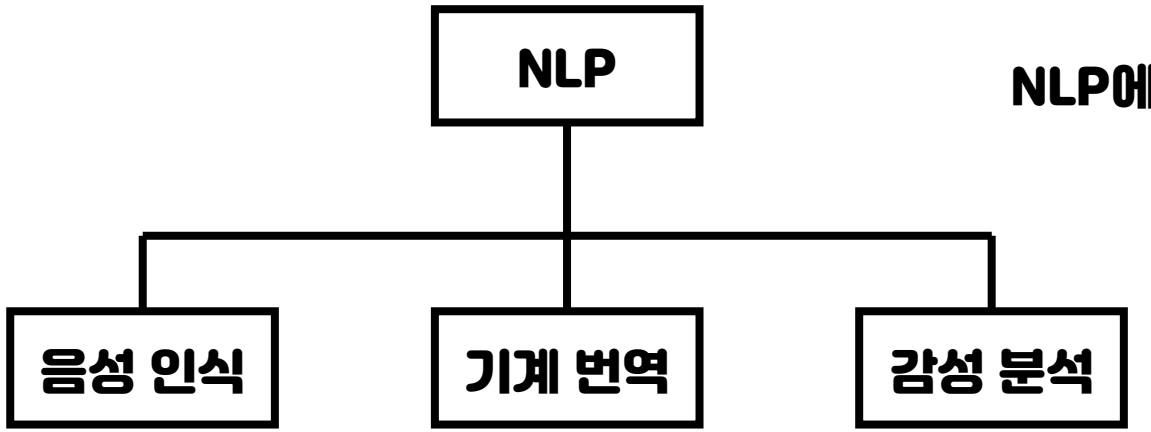
마지막 recurrent layer일 경우 마지막 timestep의 hidden state만 출력으로 내보냄

2. RNN


출력을 펼칠
필요가 없음
출력층
은닉층
20
100
샘플
(20, 100)
셀
예측
입력
(10,)
출력
소프트맥스
활성화 함수
뉴런 10개의
순환층
밀집층

- Natural Language Processing(자연어 처리)

NLP는 컴퓨터를 사용해 인간의 언어를 처리하는 분야



NLP에서는 훈련 데이터를 종종 corpus(말뭉치)로 부름

- Text Data

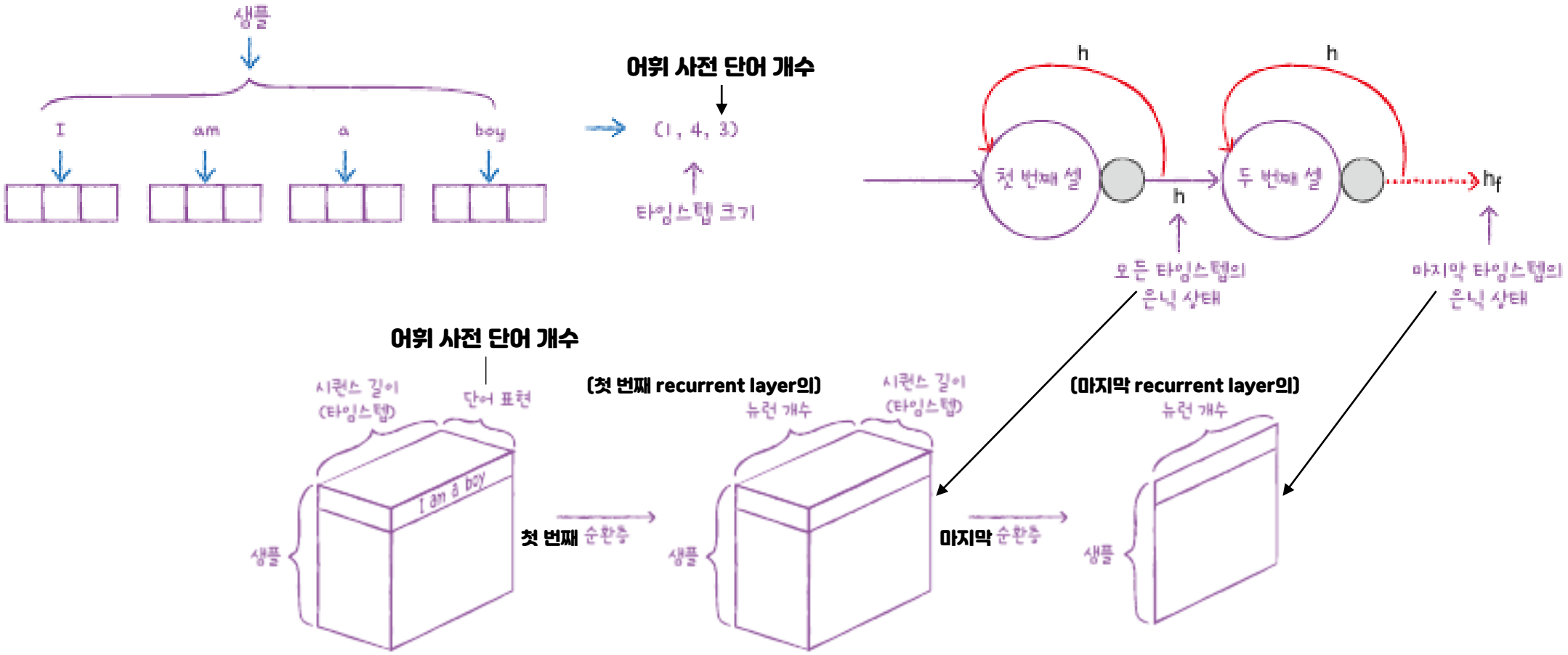
숫자가 큰 것이 더 좋은 것이라는 의미X
정수값 사이에는 어떠한 관계도 없음
token : 분리된 단어
1개의 token이 하나의 timestep

텍스트 자체를 신경망에 전달하지 않음
텍스트 데이터의 경우 **단어마다 고유한 정수**를 부여해 **숫자 데이터**로 바꿈

token에 할당하는 정수 중 몇 개는 특정한 용도로 예약되어 있음
0 : padding, 1 : 문장의 시작, 2 : 어휘 사전에 없는 token

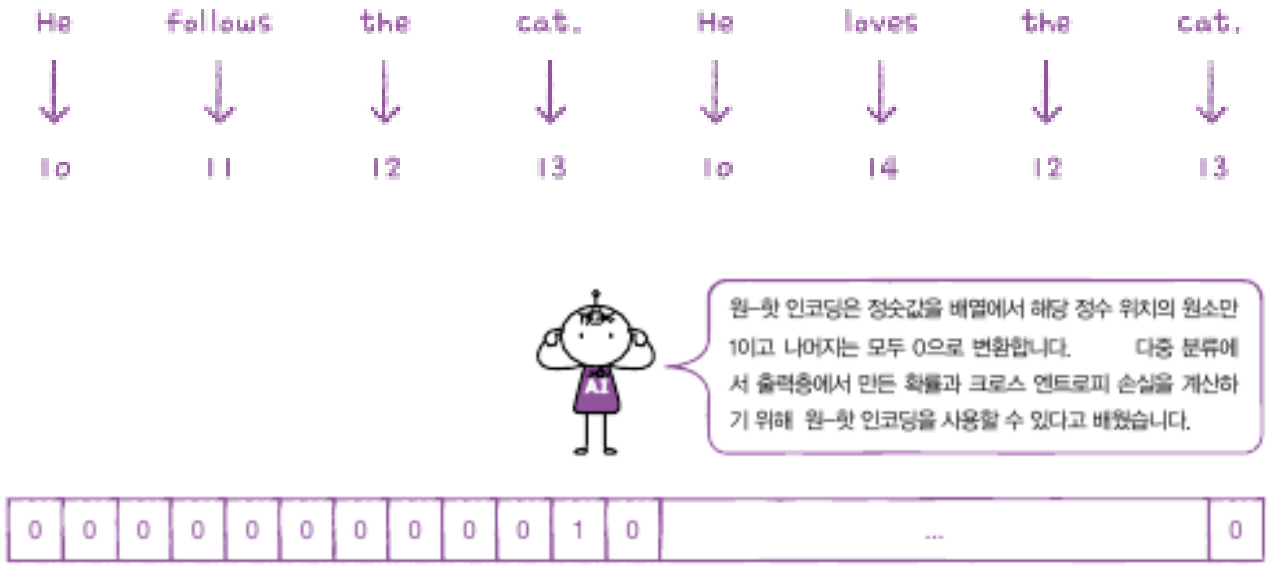
어휘 사전 : train set에서 고유한 단어를 뽑아 만든 목록
- 예) test set 안에 어휘 사전에 없는 단어가 있다면 2로 변환

2. RNN



마지막 recurrent layer일 경우 마지막 timestep의 hidden state만 출력으로 내보냄

3-1. One-hot encoding(원-핫 인코딩)



정수값 사이에는 어떠한 관계도 없기 때문에 one-hot encoding을 통해 각 숫자값에 따른 중요도를 배제할 수 있음

데이터가 매우 커지는 단점이 있음

3-2. word embedding(단어 임베딩)

'Cat'의 단어 임베딩 벡터

embedding 벡터의 크기 : 10

embedding 벡터의 크기는 하이퍼 파라미터

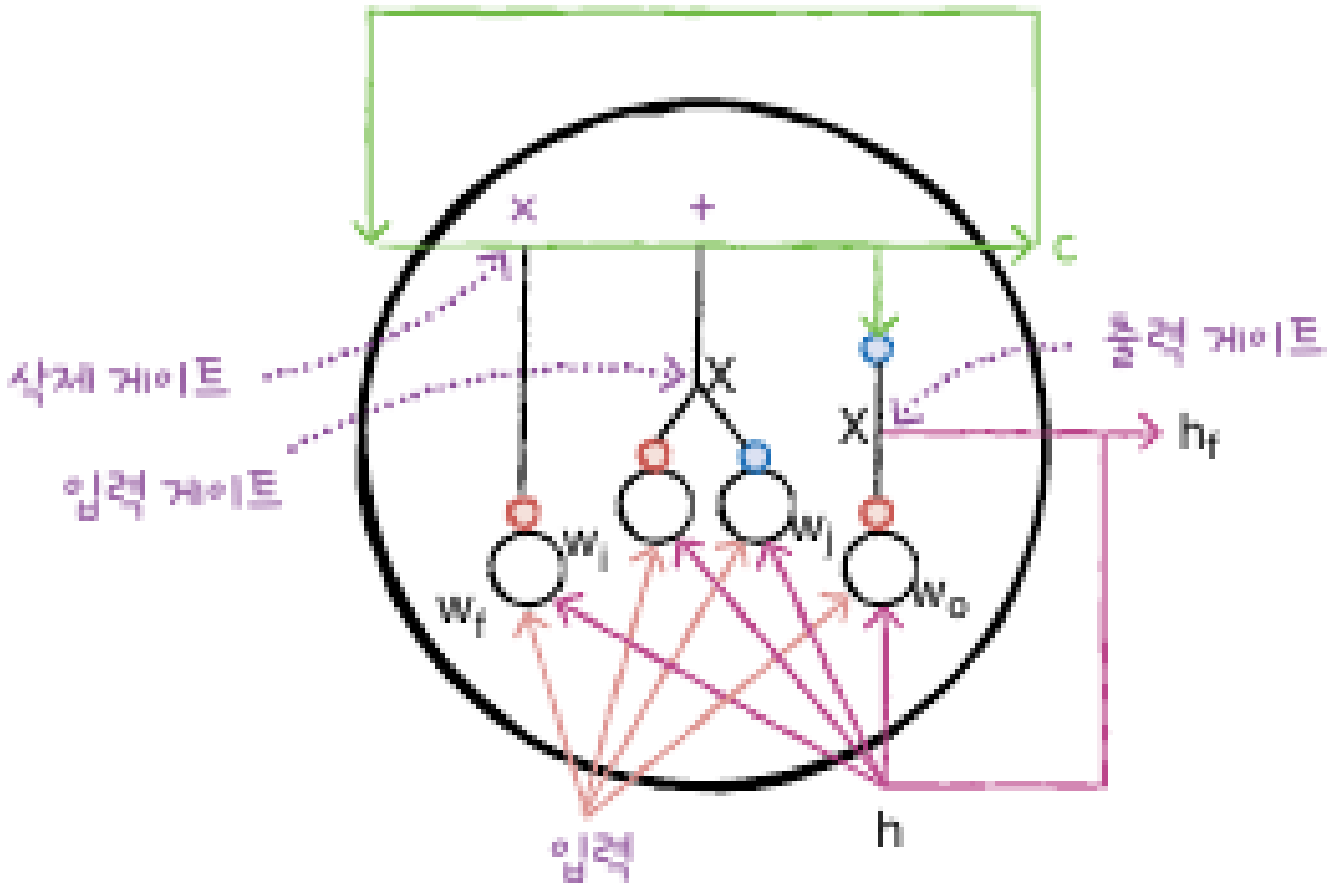
처음에는 모든 벡터가 랜덤하게 초기화되지만 훈련을 통해 좋은 word embedding을 학습

word embedding – 정수 -> 실수 밀집 벡터

밀집 벡터는 단어 사이의 관계를 표현 가능

word embedding으로 만들어진 벡터는 one-hot encoding된 벡터보다
훨씬 의미있는 값으로 채워져 있어 NLP에서 더 좋은 성능을 내는 경우가 많음

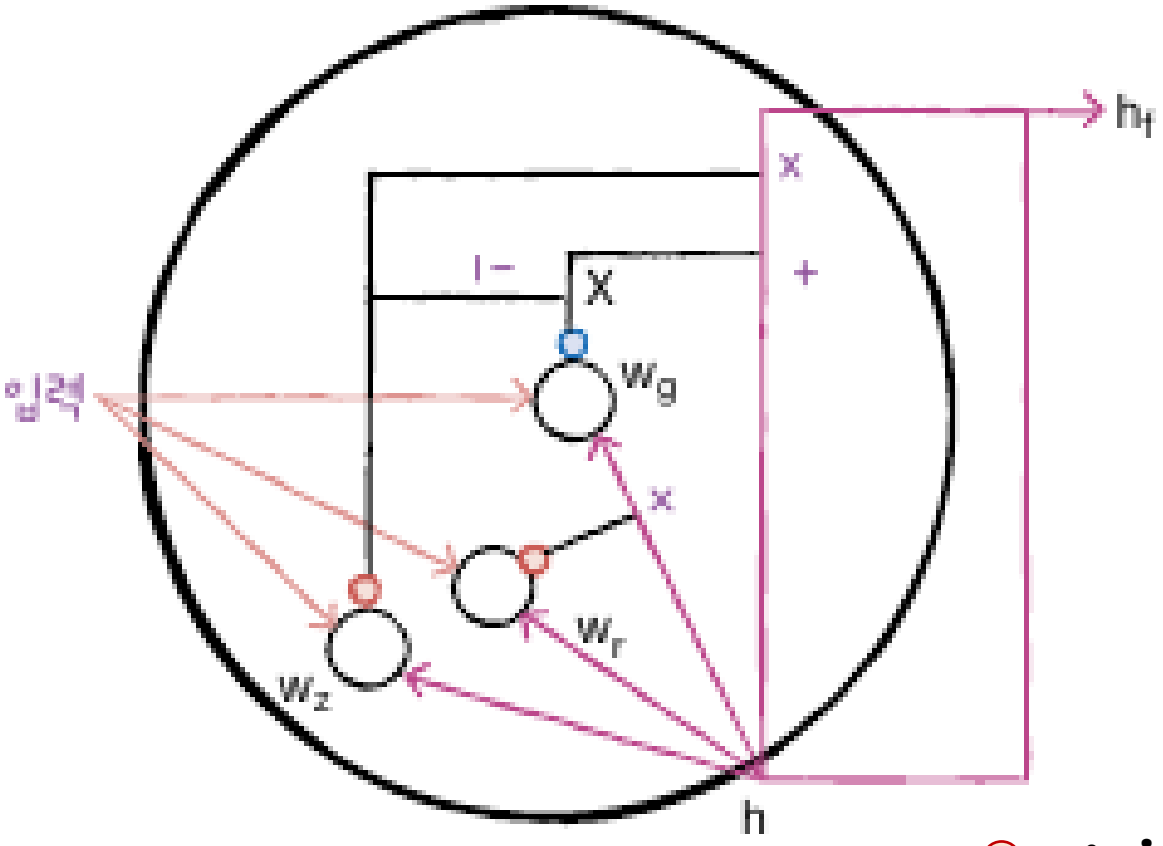
4. Long Short-Term Memory(LSTM)



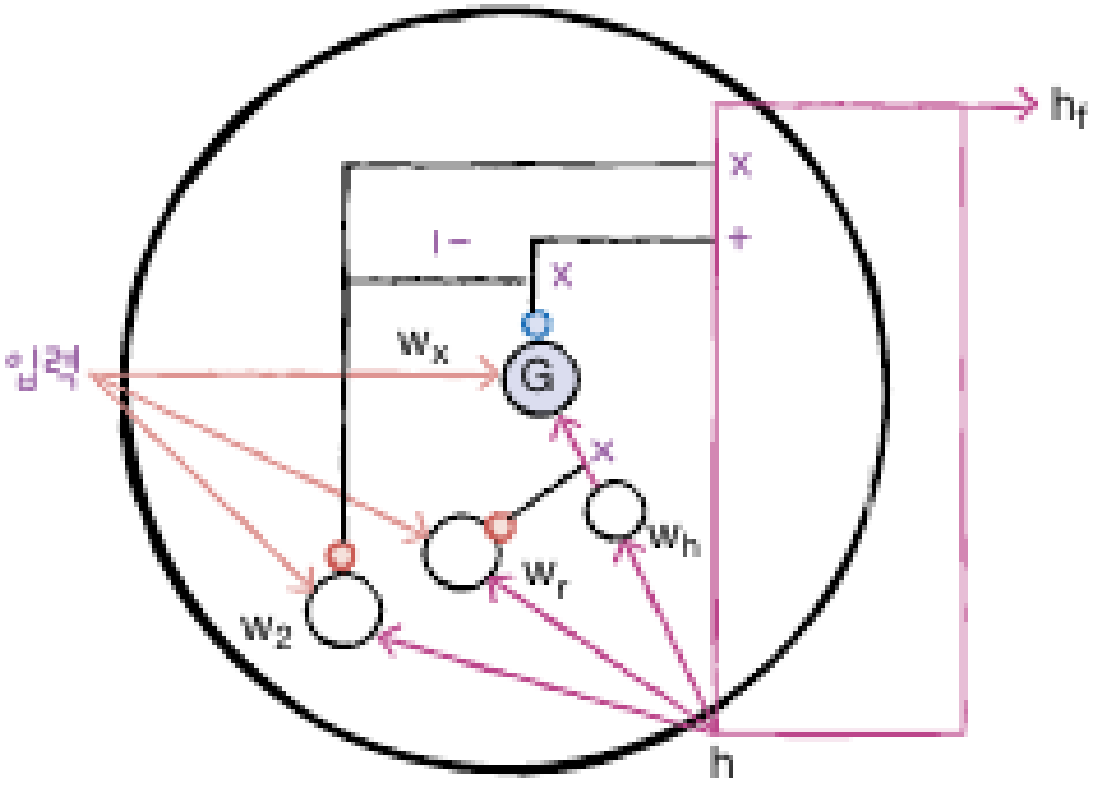
○ : sigmoid 함수

○ : tanh 함수

4. Gated Recurrent Unit(GRU)



TensorFlow에 구현된 GRU



○ : sigmoid 함수

○ : tanh 함수